



VHDL/Verilog-HDL テストベンチ・サンプル記述集 (後編)

鳥海佳孝, 田原迫仁治, 横溝憲治

いよいよテストベンチの連載も今回が最終回です。これまでの内容から、テストベンチをきちんと作ることがいかに重要で、また実際に“よい”テスト・パターンを作ることがいかにむずかしいかがおわかりいただけたと思います。機能検証を行う方法やツールにはいろいろなものがありますが、今回の前半では、そのなかでも波形表示ツールの有効性について、筆者の体験を交えながらお話ししていきたいと思います。後半では、いつものように実際の設計に役立つテストベンチのサンプル記述を紹介しします。なお、本記事で紹介する記述はModelSim, VeriLogger Proなどで動作を確認しています。また、各サンプル記述は本誌付属のCD-ROMに収録されています。

1. デバッグの極意

——波形表示ツールの活用術

さて、機能検証というのは基本的に、これから作ろうとしているLSIなりシステムなりの入力に、適切な信号パターンを入力して、そこから出力された信号などを観測します。一番簡単で、イメージのわきやすい観測方法として、波形を表示させる方法があります。現在入手可能な波形表示ツールには、以下の2種類があります。

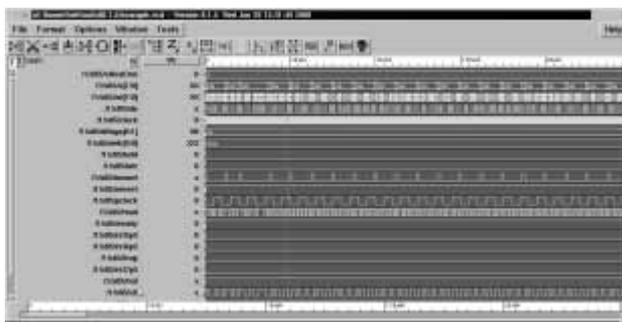
- (1) HDL シミュレータに付属しているもの
- (2) 単体で販売されており、VCD (Value Change Dump)

ファイルを読み込めるようになっているもの

このうち(1)のほうは、場合によっては(ほとんどの場合?)それぞれの波形情報がツール・ベンダの独自フォーマットになっているため、異なるベンダのツールの間で波形情報をやり取りすることがなかなかむずかしい状況となっています。一方、(2)では、VCDファイルという、共通のテキスト・フォーマットにのっとって出力されたファイルを読み込んで表示させるため、VCDファイルを読み込めるツールであれば相互に波形情報を交換することができます。VCDファイルの波形情報を表示できるツールとしてはさまざまなものがありますが、たとえば筆者は以下の製品を使用することがあります。

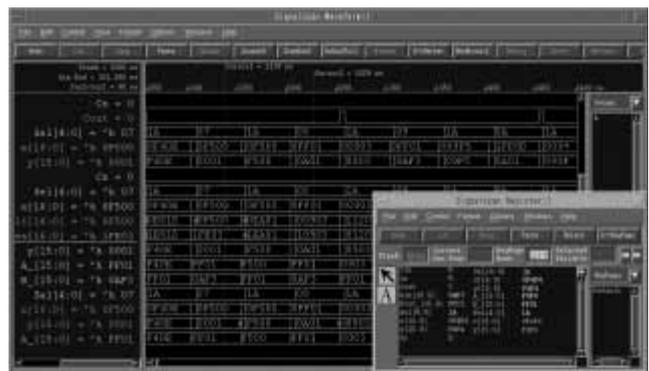
- 米国 Veritools 社の「Undertow」(図1)
- 米国 Cadence Design Systems 社の「DAI Signalscan」(図2)
- 米国 SynaptiCAD 社の「Timing Diagrammer Pro」(図3)

このようなVCDファイルをサポートしている波形表示ツールがあれば、じつは手元にHDLシミュレータがなくても、ある程度のデバッグが可能になります。つまり、VCDファイルとその波形情報を出力したHDLソース(RTLデータとテストベンチ)があれば、シミュレータを動作させることなくデバッ



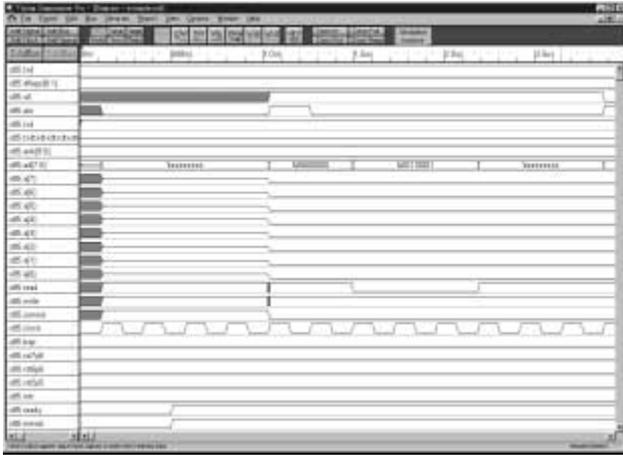
〔図1〕 Undertowの波形表示

米国 Veritools 社の製品。ツールに添付されているサンプルのVCDファイルを読み込んでいるようす。



〔図2〕 Signalscanの波形表示

米国 Cadence Design Systems 社の製品。開発したのは米国 Design Acceleration 社 (DAI) だが、1999年に同社はCadence社によって買収された。



【図3】 Timing Diagrammer Pro の波形表示

米国 SynaptiCAD 社の製品。このツールはVCDだけでなく、いろいろなフォーマットのファイルをサポートしている。

ができるのです。

実際、筆者はHDLシミュレータなしに、波形表示ツールだけを使用してデバッグした体験があります。ある顧客から、以前に筆者が手がけたVerilog-HDLソースを、できる限り早く修正してほしいという依頼がありました。それを行うにあたって、以下のようないくつかの問題がありました。

- 筆者の自宅から顧客のところまで、距離的に遠かった(移動する時間とコストがもったいない)
- HDLソースを丸ごと送られても、筆者の手元にまともなVerilog-HDLシミュレータがなかった
- 長年の経験から、他人の構築したテスト環境を手元で再現するのは、かなりの時間を要することがわかっていた

このような状況に対処するために、筆者は次のような手順を考えました(図4)。

- (1) 先方からHDLソース(RTL, テストベンチ)を電子メールで送ってもらう
- (2) 不具合の出ているVCDファイル(波形データ)を電子メールで送ってもらう
- (3) VCDファイルを筆者の自宅の波形表示ツールで解析する
- (4) HDLソースを修正し、先方に電子メールで送る
- (5) 先方でシミュレーションを実行してもらう
- (6) (2)以降をバグがなくなるまで繰り返す

基本的に不具合箇所は顧客のほうで絞り込んでいただいたので、VCDファイルは、不具合の影響が及んでいるブロックに関連する部分だけを出力してもらいました(さすがに全ダンプを送ってもらうとなると、筆者の通信環境ではちょっと厳しい。ディスクの容量はなんとでもなるのだが…)。

シミュレーションは顧客側のエンジニアにお願いしているので、さすがに、「試しにこの信号の論理を変更して、ちょ



(すべて電子メールでやりとり)

【図4】 波形表示ツールによるデバッグの手順

顧客とVCDファイルなどを電子メールによってやり取りした。

っとシミュレーションで流してみようか」というわけにはいきません。HDLの修正はできるだけ慎重に行いました。結局、丸々2日間、合計6度の修正を試みて、なんとかバグをフィックスさせました。いまにして思えば、本当は4度くらいの修正でなんとかなった気もするのですが、手元に適当なHDLシミュレータがなかったため、顧客側のエンジニアの方に余計な手間をかけていただくことになってしまいました。

しかし、このアプローチは結果的によかったと筆者は思っています。なぜなら、手元にHDLシミュレータがないので、修正に際して、じっくりといろいろなことを考えざるを得なかったからです(ちょうど将棋で、次に指す一手について長考しているようなもの)。そのおかげで、普段であればここまで考えないだろうというところまで、深く考えました。そのときに、普段、いかにむだなシミュレーション(いわゆるトライ・アンド・エラーのシミュレーション)を行っているかということに認識させられました。上述のような環境でデバッグを行う場合、HDLの文法ミスなどというつまらないエラーは言語道断です。できる限り自分の目でチェックして、そのようなくだらないミスをなくさなければ、デバッグの効率がものすごく悪くなってしまいます。

波形表示ツールが世の中に登場したころは、HDLシミュレータにも付属しているのに、なぜ独立したツールが必要になるのか、筆者は非常に不思議に思っていました。しかし高価なHDLシミュレータが手元になくても、上記のような方法でデバッグして仕事をこなせるわけですから、波形表示ツールの存在価値はかなりのものだ、いまでは認識しています。しかも、値段的にもHDLシミュレータに比べれば、非常に安価です。

以上のことを考えると、個人的に、またはある程度対話的にシミュレーションやデバッグを行いたい場合には、WindowsやLinuxなどの上で動作する安価な(または、無償の)HDLシミュレータを利用すればよいと思います¹⁾。一